

Τμήμα Μηχανικών Η/Υ και Πληροφορικής Πανεπιστημίου Ιωαννίνων
ΜΥΕ047: Αλγόριθμοι για Δεδομένα Ευρείας Κλίμακας
Ακαδημαϊκό Έτος 2021-22
Διδάσκων: Σπύρος Κοντογιάννης

2ο Σετ Ασκήσεων: Ανίχνευση Κοντινότερων Γειτόνων σε Συλλογή Εγγράφων

Ανακοίνωση: Τετάρτη, 25 Μαΐου 2022

Παράδοση: Δευτέρα, 27 Ιουνίου 2022

Τελευταία Ενημέρωση: Τετάρτη, 25 Μαΐου 2022

1. ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΣΙΑΣ

Στόχος της παρούσας εργασίας είναι η ανίχνευση κοινοτήτων σε γραφήματα κοινωνικών δικτύων, μέσω τεχνικών συσταδοποίησης. Θα πρέπει να υλοποιηθούν ιεραρχικές τεχνικές συσταδοποίησης, και να υπολογιστεί η «βέλτιστη» διαμέριση του γραφήματος σε κοινότητες (δεδομένης της επιλεχθείσας ιεραρχικής συσταδοποίησης), βάσει της μετρικής αρθρωτότητας (modularity).

Θα πρέπει επίσης να πειραματιστείτε με τους αλγορίθμους που υλοποιήσατε, σε ένα πραγματικό στιγμιότυπο κοινωνικού δικτύου. Ζητούμενο είναι να γίνει σύγκριση των διαφορετικών τεχνικών ανίχνευσης κοινοτήτων, τόσο ως προς την ταχύτητα εκτέλεσης όσο και ως προς την ποιότητα των παρεχόμενων διαμερίσεων των κορυφών του γραφήματος σε μη-επικαλυπτόμενες κοινότητες.

Απαραίτητες βιβλιοθήκες:

- Networkx ($\geq v.2.8.1$),
- Matplotlib ($\geq v.3.5.2$),
- Numpy ($\geq v.1.22$),
- Pandas ($\geq v1.4.2$)

(1α) Ανάγνωση & Επεξεργασία Δεδομένων Εισόδου

Το στιγμιότυπο με το οποίο θα εργαστείτε δίνεται μέσα από τα εξής σύνολα δεδομένων:

- **fb-pages-food.nodes**

Το συγκεκριμένο (comma-separated-values) αρχείο παραθέτει λεπτομέρειες για τις οντότητες του κοινωνικού δικτύου. Έχει την εξής δομή:

```
=====
```

id,	name,	new_id
402449106435352,	Josh Marks,	386
368969274888,	Blue Ribbon Restaurants,	473
765596333518863,	Pat Neely,	1
...		

```
=====
```

Η πρώτη στήλη αφορά μοναδικά αναγνωριστικά των κόμβων, η δεύτερη στήλη παραθέτει τα πλήρη ονόματα των οντοτήτων που αντιστοιχούν στους κόμβους, ενώ η τρίτη στήλη παραθέτει νέα αναγνωριστικά, από το σύνολο $\{0,1,2, \dots, 619\}$ για όλους τους (620 στο πλήθος τους) κόμβους του γραφήματος, για λόγους καλύτερης αναγνωσιμότητας του γραφήματος που προκύπτει.

- **fb-pages-food.edges**

Το (επίσης comma-separated-values) αρχείο παραθέτει όλες τις ακμές του γραφήματος, μία ακμή σε κάθε γραμμή του. Έχει λοιπόν την εξής δομή:

```
=====
```

node_1,	node_2
0,	276
0,	58
0,	132
0,	603
0,	398

```
=====
```

0,	555
1,	265
1,	611
2,	265
2,	182
2,	345
...	

Η πρώτη στήλη (node_1) παραθέτει το αναγνωριστικό (new_id) της ουράς της εκάστοτε ακμής, ενώ η δεύτερη στήλη (node_2) παραθέτει το αναγνωριστικό (new_id) της κεφαλής της. Πχ, η συνολική λίστα των ακμών του γραφήματος θα έπρεπε να φαίνεται ως εξής:

[(0,276), (0,58), (0,132), (0,603), (0,398), (0,555), (1,265), (1,611), (2,265), (2,182), (2,345), ...]

Θα πρέπει να κάνετε ανάγνωση των δεδομένων του γραφήματος (ουσιαστικά, των ακμών από το σύνολο **fb-pages-food.edges**) σε ένα μητρώο (με όνομα **fb_links**), τύπου dataframe της βιβλιοθήκης **pandas**, με δύο στήλες με ονόματα node_1 και node_2. Δείτε για παράδειγμα την εντολή `read_csv(...)` της βιβλιοθήκης **pandas**.

Για λόγους ευκολότερου πειραματισμού, δώστε τη δυνατότητα στον χρήστη να καθορίζει υποσύνολο των πρώτων στη σειρά **MAX_NUM_LINKS** ακμών από το **fb_links** που θα χρησιμοποιηθούν στο υπό κατασκευή γράφημα. Αποθηκεύστε σε ένα νέο dataframe **fb_links_df** με αυτές τις **MAX_NUM_LINKS** ακμές. Στο εξής θα θεωρήσετε μόνο αυτό το υποσύνολο ακμών, αλλά και τις κορυφές που βρίσκονται στα άκρα τους.

Στη συνέχεια, θα πρέπει να κατασκευάσετε το (υπο)γράφημα του κοινωνικού δικτύου που περιγράφεται όλες τις ακμές (και μόνο αυτές) που αποθηκεύσατε στο **fb_links_df**, μαζί με τις κορυφές (και μόνο αυτές) που βρίσκονται στα άκρα των ακμών. Πρόκειται, δηλαδή, για το υπογράφημα του κοινωνικού μας δικτύου που επάγεται από το συγκεκριμένο υποσύνολο ακμών που αποθηκεύετε.

Η δημιουργία (και μετέπειτα επεξεργασία) του γραφήματος θα πρέπει να γίνει μέσω της βιβλιοθήκης **network**. Το γράφημα που θα κατασκευάσετε πρέπει να είναι ακατεύθυντο (undirected), καθώς οι τεχνικές ανίχνευσης κοινοτήτων με τις οποίες θα ασχοληθούμε εφαρμόζονται σε ακατεύθυντα γραφήματα. Για να το κάνετε αυτό, μπορείτε για παράδειγμα να κάνετε χρήση της ακόλουθης εντολής:

```
G = nx.from_pandas_edgelist(fb_links_df, "node_1", "node_2", create_using=nx.Graph())
```

Το τελευταίο όρισμα της εντολής αυτής ορίζει ότι θα γίνει χρήση της μεθόδου **nx.Graph()**, η οποία δημιουργεί ένα ακατεύθυντο γράφημα από το dataframe **fb_links_df** που περιγράφει όλες τις ακμές που θέλουμε να λάβουμε υπόψη μας. Τέλος, θα πρέπει να γίνει κατάλληλη χρήση της βιβλιοθήκης **matplotlib.pyplot** για την αποτύπωση ενός γραφήματος στην οθόνη.

PROGRAMMING TASK 1A:

(i) Δημιουργήστε ρουτίνα `STUDENT_AM_read_graph_from_csv(...)`, η οποία διαβάζει ολόκληρο το CSV αρχείο ακμών, **fb-pages-food.edges**, και δημιουργεί ένα pandas dataframe **MONO** για τις **NUM_LINKS** ακμές, με το όνομα **fb_links_df**. Η ρουτίνα θα πρέπει να διαγράφει όλες τις ακμές (δλδ, γραμμές) από το dataframe που αφορούν ακμές-βρόχους. Το τελικό dataframe ακμών θα πρέπει να είναι το **fb_links_loopless_df**. Τέλος, από το dataframe με τις ακμές που απέμειναν, θα πρέπει να δημιουργηθεί το κατάλληλο (ακατεύθυντο) γράφημα, που επιστρέφεται από τη συγκεκριμένη ρουτίνα, μαζί με τη λίστα `node_names_list` ονομάτων των κόμβων του.

(ii) Χρησιμοποιήστε κατά το δοκούν την (έτοιμη στο TEMPLATE) ρουτίνα εκτύπωσης γραφημάτων, `my_graph_plot_routine()`, για την αποτύπωση του γραφήματος στην οθόνη σας.

(1β) Εμπλουτισμός Γραφήματος

Σε αυτό το βήμα ζητείται η παροχή δυνατότητας επαύξησης του γραφήματος με επιπρόσθετες ακμές, έτσι ώστε είτε να μειωθούν οι συνεκτικές του συνιστώσες, είτε να αποφευχθεί η ύπαρξη γεφυρών (πχ, εμφυτεύοντας έναν κύκλο ΧΑΜΙΛΤΟΝ, που περιλαμβάνει όλες τις κορυφές του γραφήματος).

- Η πρώτη ρουτίνα θα επιτρέπει την προσθήκη, από κάθε κόμβο X ξεχωριστά, ενός πλήθους `NUM_RANDOM_EDGES` ακμών προς τυχαίους κόμβους Y (η επιλογή γίνεται μεταξύ όλων των ΜΗ ΓΕΙΤΟΝΩΝ του X). Για κάθε Y , θα αποφασίζεται, με ρίψη τυχαίου νομίσματος το οποίο έχει πιθανότητα επιτυχίας `EDGE_ADDITION_PROBABILITY`, αν τελικά θα προστεθεί στο γράφημα η ακμή XY . Η ρουτίνα επιστρέφει το επαυξημένο γράφημα, μετά την επεξεργασία όλων των κορυφών X του γραφήματος εισόδου.
- Η δεύτερη ρουτίνα θα επιτρέπει την «εμφύτευση» στο γράφημα ενός κύκλου Χάμιλτον, δηλαδή, ενός κύκλου που περιλαμβάνει όλες τις κορυφές του γραφήματος. Δέχεται ως είσοδο (το γράφημα και) μια λίστα `node_names_list` με τα ονόματα (αναγνωριστικά) των κορυφών. Εμφυτεύει τον κύκλο Χάμιλτον που επισκέπτεται τις κορυφές του γραφήματος με τη σειρά που ορίζει η `node_names_list`.

PROGRAMMING TASK 1B:

(i) Δημιουργήστε τη ρουτίνα `STUDENT_AM_add_random_edges_to_graph(...)`

(ii) Δημιουργήστε τη ρουτίνα `STUDENT_AM_add_hamilton_cycle_to_graph(...)`

(1γ) Δημιουργία Διμερούς Συσταδοποίησης Γραφήματος

Σε αυτό το βήμα καλείστε να δημιουργήσετε ρουτίνα προσδιορισμού κοινοτήτων στο γράφημα εισόδου G , χρησιμοποιώντας τη λογική του αλγορίθμου GIRVAN-NEWMAN (GN) που είδαμε στο μάθημα. Θα ασχοληθείτε με μια συγκεκριμένη υλοποίηση, η οποία:

- Αρχικά θα υπολογίζει τις $K \geq 1$ συνεκτικές συνιστώσες του γραφήματος, και δημιουργεί μια λίστα κοινοτήτων (κάθε κοινότητα είναι λίστα κόμβων) όπου κάθε συνιστώσα εκλαμβάνεται ως μία διαφορετική κοινότητα.
- Στη συνέχεια, θα εφαρμόζει τη λογική του αλγορίθμου GN στη μεγαλύτερη συνεκτική συνιστώσα (GCC) του γραφήματος, που αντιστοιχεί στην κοινότητα (λίστα κόμβων) LC , αφαιρώντας από αυτή διαδοχικά ακμές (μία προς μία) με τη μεγαλύτερη τιμή *betweenness-centrality*, μέχρι την πρώτη φορά που αποσυνδέεται η συγκεκριμένη συνιστώσα (σε ακριβώς δύο συνιστώσες). Οι δυο αυτές συνιστώσες υποδεικνύουν δυο νέες υπο-κοινότητες, έστω $LC1$ και $LC2$.
- Τέλος, θα αφαιρεί από την `community_tuples` την κοινότητα LC , και θα προσθέτει τις κοινότητες $LC1$ και $LC2$.

Η `networkx` έχει μια έτοιμη ρουτίνα, `girvan_newman`, που υλοποιεί ακριβώς το συγκεκριμένο σκεπτικό. Θα πρέπει σε πρώτη φάση να χρησιμοποιήσετε την έτοιμη συνάρτηση βιβλιοθήκης, και στη συνέχεια να δώσετε και τη δική σας υλοποίηση.

PROGRAMMING TASK 1Γ:

(i) Δημιουργήστε τη ρουτίνα `STUDENT_AM_use_nx_girvan_newman_for_communities(...)` που αξιοποιεί τη συνάρτηση βιβλιοθήκης `girvan_newman` της `networkx` για τη δημιουργία $K+1$ κοινοτήτων, σε ένα γράφημα με K συνεκτικές συνιστώσες.

(ii) Δημιουργήστε τη δική σας ρουτίνα `STUDENT_AM_one_shot_girvan_newman_for_communities(...)` που εντοπίζει $K+1$ κοινότητες, σε ένα γράφημα με K συνεκτικές συνιστώσες, σύμφωνα με τα προαναφερθέντα.

(18) Δημιουργία Διμερούς Συσταδοποίησης Γραφήματος

Υπάρχουν πολλοί τρόποι δημιουργίας μιας ιεραρχίας διαμερίσεων του γραφήματος σε κοινότητες. Θα ασχοληθούμε εδώ με μια συγκεκριμένη μέθοδο, τη **Διαιρετική Συσταδοποίηση** (divisive clustering). Πρόκειται για μια μέθοδο κατασκευής ιεραρχικής συσταδοποίησης, «από πάνω προς τα κάτω», η οποία εκκινεί με ακριβώς K κοινότητες, μία για κάθε συνεκτική συνιστώσα του γραφήματος, και στη συνέχεια διαρκώς υποδιαιρεί τη μέγιστη κοινότητα σε δύο υποκοινότητες σύμφωνα με τον αλγόριθμο που δημιουργήσατε στο (1γ). Η διαδικασία αυτή τερματίζεται όταν θα έχουν δημιουργηθεί ακριβώς num_divisions κοινότητες στο γράφημα.

Όμως, κατά τη διαδικασία αυτή θα πρέπει να αποτυπωθούν όλες οι διαμερίσεις (από K κοινότητες, έως num_divisions κοινότητες). Για να γίνει αυτό, προτείνεται η συνάρτηση να επιστρέφει

(ι) την αρχική διαμέριση του γραφήματος σε K κοινότητες (μία για κάθε συνιστώσα του).

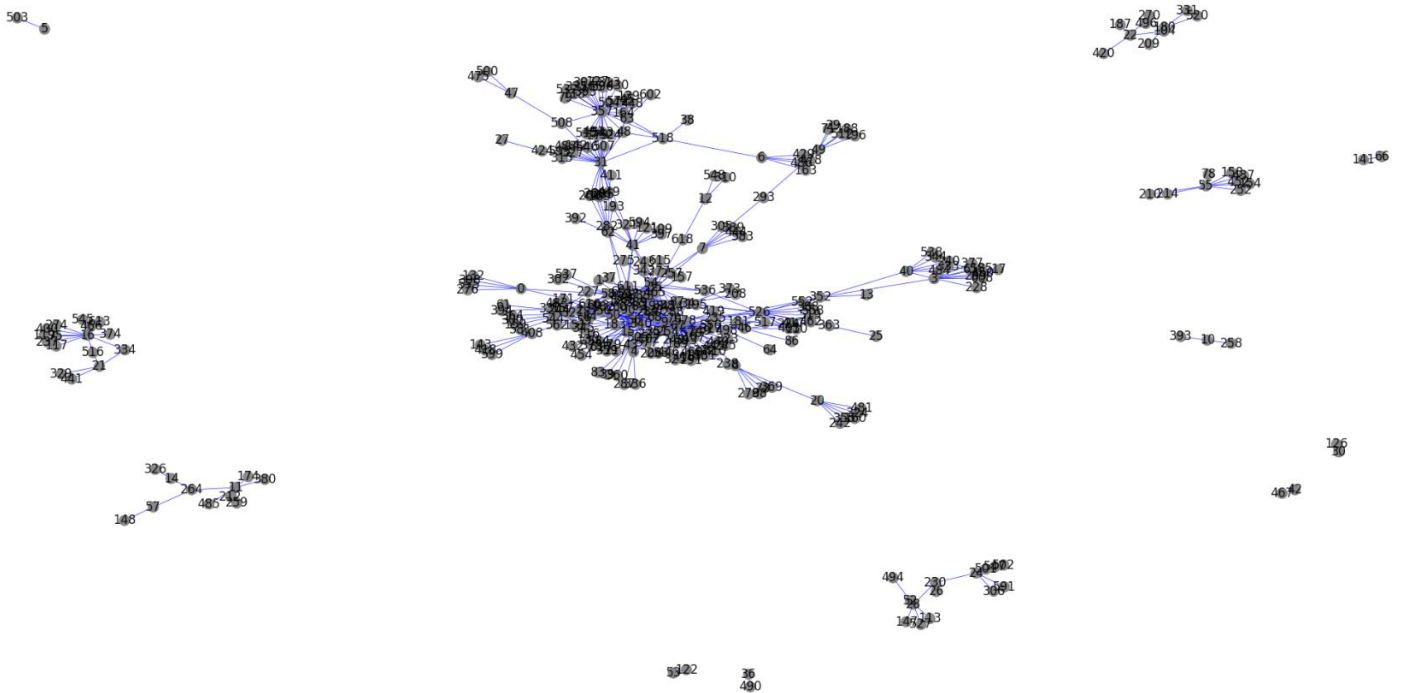
(ιι) Μια λίστα περιλαμβάνει τριάδες από tuples (μία για κάθε κοινότητα), όπου, σε κάθε επανάληψη θα αποθηκεύεται η τριάδα $[LC, LC1, LC2]$ για να γνωρίζουμε ότι η επόμενη στη σειρά διαμέριση, έστω NEXT_PARTITION , προκύπτει από την τρέχουσα διαμέριση, έστω CUR_PARTITION , ως εξής:

- Αφαιρείται από την CUR_PARTITION η LC .
- Προστίθεται στην CUR_PARTITION η $LC1$ και η $LC2$.

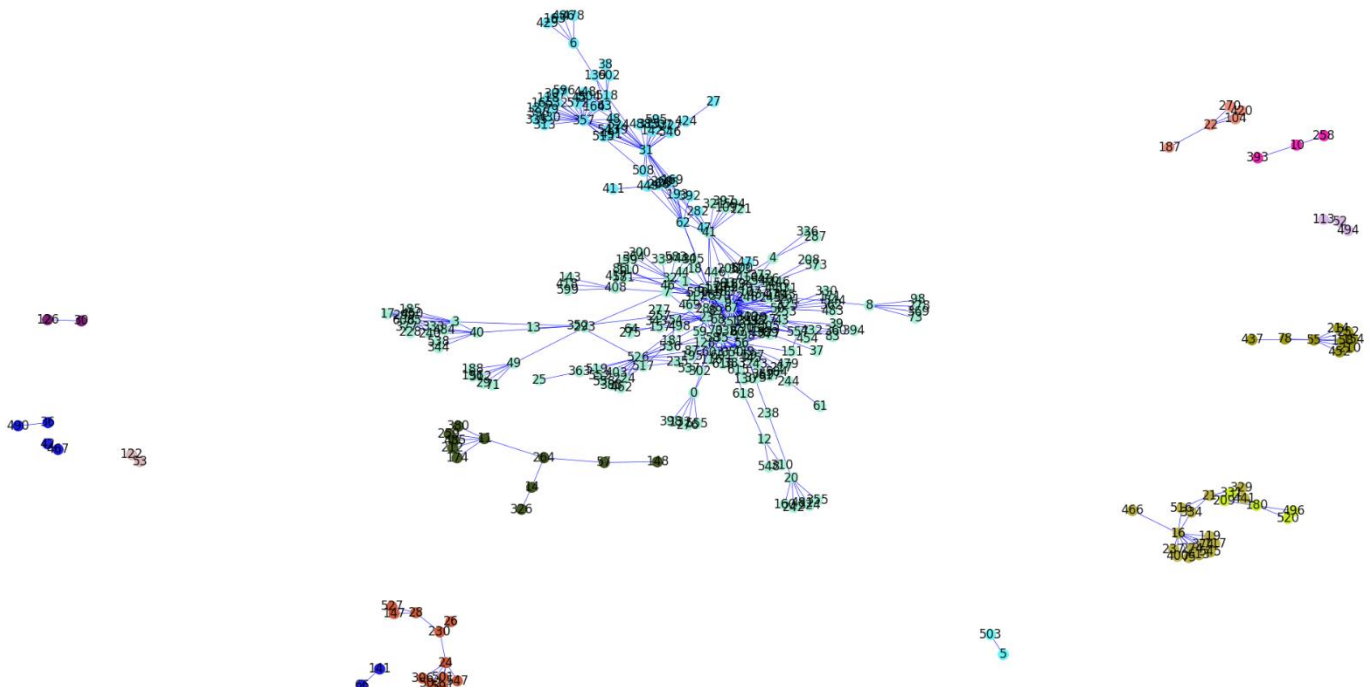
Διατίθενται ορισμένα έτοιμα εργαλεία της **networkx**, τα οποία μπορείτε να αξιοποιήσετε κατά το δοκούν:

- Ρουτίνα **connected_components** (βλ. εγχειρίδιο `networkx.2.8.1`, ενότητα 3.15.1, σελ.253): Υπολογίζει τις συνεκτικές συνιστώσες ενός ακατεύθυντου γραφήματος.
- Ρουτίνα **edge_betweenness_centrality** (βλ. εγχειρίδιο `networkx.2.8.1`, ενότητα 3.7.5, σελ.253): Υπολογίζει τις *betweenness* τιμές όλων των ακμών, ως προς όλες τις κορυφές-ρίζες.
- Ρουτίνα **edge_betweenness_centrality_subset** (βλ. εγχειρίδιο `networkx.2.8.1`, ενότητα 3.7.5, σελ.254): Υπολογίζει τις *betweenness* τιμές όλων των ακμών, ως προς ένα συγκεκριμένο (πχ, τυχαία επιλεγμένο) υποσύνολο από κορυφές-ρίζες.
- Ρουτίνα **girvan_newman** (βλ. εγχειρίδιο `networkx.2.8.1`, ενότητα 3.14.9, σελ.317): Υλοποίηση του αλγορίθμου Girvan-Newman.

Έστω ότι μελετάμε, στο δικό μας σύνολο δεδομένων, το ακατεύθυντο γράφημα που προκύπτει από τις πρώτες 500 ακμές (499, μετά τη διαγραφή βρόχων). Η ακόλουθη εικόνα παρουσιάζει το συγκεκριμένο γράφημα, το οποίο περιλαμβάνει 15 συνεκτικές συνιστώσες:



Ο αλγόριθμος `giran-newman` της βιβλιοθήκης **networkx** ουσιαστικά θεωρεί κάθε συνιστώσα, από μόνη της, ως μία κοινότητα. Επίσης, υποδιαιρεί και τη μέγιστη συνεκτική συνιστώσα (κοινότητα) σε δυο υπο-κοινότητες από κάποια αρχική συνιστώσα, που προκύπτουν με διαδοχικές αφαιρέσεις ακμών μέγιστης `betweenness`-τιμής. Στο συγκεκριμένο γράφημα, λοιπόν, θα προκύψουν 16 κοινότητες, αφού η μεγαλύτερη συνιστώσα υποδιαιρείται σε δύο κοινότητες. Το αποτέλεσμα που θα προκύψει, για το συγκεκριμένο γράφημα των 500 πρώτων ακμών από το σύνολο δεδομένων, φαίνεται στο ακόλουθο σχήμα:



Ο συγκεκριμένος αλγόριθμος βιβλιοθήκης παράγει πολύ φτωχό αποτέλεσμα σε αρκετές περιπτώσεις, πχ σε γραφήματα δίχως γέφυρες.

PROGRAMMING TASK 1Δ:

Δημιουργήστε ρουτίνα `STUDENT_AM_divisive_community_detection(...)` η οποία θα δημιουργεί διαιρετικά την ιεραρχία διαμερίσεων του γραφήματος, από μια διαμέριση K κοινοτήτων μέχρι την τελική διαμέριση `num_divisions` κοινοτήτων. Δώστε στον χρήστη τη δυνατότητα να υπολογίζει τις Edge-Betweenness τιμές είτε επακριβώς, είτε προσεγγιστικά χρησιμοποιώντας μόνο ένα μικρό υποσύνολο κορυφών (πχ, το 10% των κόμβων) ως ρίζες των BFS δένδρων. Η ρουτίνα θα επιστρέφει ολόκληρη την ιεραρχία διαμερίσεων.

(1Ε) Αποτύπωση Διαμέρισης Γραφήματος σε κοινότητες

Σε αυτό το τελικό βήμα ζητείται να δοθούν δύο διαφορετικές ρουτίνες. Η πρώτη θα αποτυπώνει συγκεκριμένη διαμέριση του γραφήματος σε κοινότητες, ενώ η δεύτερη θα μετρά, για κάθε διαμέριση στην ιεραρχία χωριστά, την τιμή αρθρωτότητας που προκύπτει για κάθε διαμέριση στην ιεραρχία. Στο τέλος θα επιστρέφεται γραφική παράσταση με τις τιμές αρθρωτότητας ανά διαμέριση (δηλαδή, για τις τιμές $\{ K, K+1, \dots, \text{num_divisions} \}$), ενώ θα δίνεται δυνατότητα σχεδίασης συγκεκριμένης διαμέρισης, με ένα αυθαίρετο πλήθος διαμερίσεων από το $\{ K, K+1, \dots, \text{num_divisions} \}$.

PROGRAMMING TASK 1Ε:

(ι) Δημιουργήστε ρουτίνα `STUDENT_AM_visualize_communities(...)` η οποία θα δέχεται ως είσοδο μια λίστα κοινοτήτων, θα αναθέτει τυχαία χρώματα και δημιουργεί τυχαία χρώματα για τους κόμβους (ένα χρώμα ανά κοινότητα), και αποτυπώνει στη συνέχεια το γράφημα.

(ιι) Δημιουργήστε ρουτίνα `STUDENT_AM_determine_opt_community_structure(...)` η οποία θα δέχεται ως είσοδο την ιεραρχία διαμερίσεων σε κοινότητες, και για κάθε διαμέριση θα υπολογίζει την τιμή αρθρωτότητας (modularity value). Στη συνέχεια, θα δίνει ραβδόγραμμα με τις τιμές αρθρωτότητας για κάθε διαμέριση, ενώ θα τυπώνει και εκείνη τη διαμέριση κοινοτήτων που επιτυγχάνει τη βέλτιστη τιμή αρθρωτότητας.

2. ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ

Για να είναι εφικτή η αξιολόγηση της εργασίας σας, στο πρόγραμμα σε PYTHON (έκδοση $\geq 3.9.4$) που θα παραδώσετε θα πρέπει να κάνετε τα εξής:

1. Αξιοποιήστε το TEMPLATE που σας παρέχεται, στην ετικέτα FILES του καναλιού LAB-2. Εκεί θα βρείτε έτοιμες πολλές ρουτίνες, καθώς επίσης και ένα υποτυπώδες μενού αλληλεπίδρασης με τον χρήστη. Μέσα σε αυτό το TEMPLATE, θα πρέπει να παράσχετε τις δικές σας υλοποιήσεις για όλες τις ρουτίνες το όνομα των οποίων ξεκινά με STUDENT_AM. Αντί για STUDENT_AM, εσείς θα γράφετε τα δικά σας AM (ένα ή δύο, ανάλογα με το αν κάνετε την εργασία σας κατά μόνας ή σε ομάδες των δύο ατόμων).
2. Χρησιμοποιήστε (στην αρχή του κώδικά σας) κατάλληλα σχόλια για να προσδιορίστε επακριβώς τον τρόπο εκτέλεσης του προγράμματός σας, και τη θέση στην οποία (θεωρείτε ότι) βρίσκονται τα δεδομένα. Ιδανικά, καλό θα ήταν να χρησιμοποιείτε τον ίδιο κατάλογο όπου βρίσκεται ο κώδικας και για την αποθήκευση του αρχείου δεδομένων προς επεξεργασία.
3. Θα πρέπει να παράσχετε τις κατάλληλες υλοποιήσεις σας για τις εξής επιλογές των μενού χρήστη (οι υπόλοιπες επιλογές είναι ήδη υλοποιημένες):

(1.1) Create graph from fb-food data set (fb-pages-food.nodes and fb-pages-food.edges)

(2.1) Add random edges from each node

(2.2) Add hamilton cycle (if graph is not connected)

(2.4) Compute communities with GIRVAN-NEWMAN

(2.5) Compute a binary hierarchy of communities

(2.6) Compute modularity-values for all community partitions

(2.7) Visualize the communities of the graph

4. Πειραματιστείτε με τον κώδικά σας για διαφορετικά πλήθη ακμών από το σύνολο δεδομένων, καθώς και για τυχαία γραφήματα που δημιουργούνται στο βήμα:

(1.2) Create RANDOM Erdos-Renyi graph $G(n,p)$.

5. Σχολιάστε την ποιότητα των κοινοτήτων που προκύπτουν στα παραπάνω γραφήματα. Πώς επίσης επηρεάζεται η ποιότητα των παρεχόμενων κοινοτήτων όταν προσθέσουμε ένα (μικρό) υποσύνολο ακμών, ή έναν κύκλο Χάμιλτον, ιδιαίτερα όταν το αρχικό γράφημα έχει πολλές συνιστώσες ή έχει πολλές γέφυρες (δλδ, ακμές που δεν περιλαμβάνονται σε κύκλους)? Πώς εξηγείτε αυτά που βλέπετε?

3. ΠΑΡΑΔΟΣΗ ΕΡΓΑΣΙΑΣ

Δημιουργήστε τα εξής αρχεία:

(α) ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ: Ένα συμπιεσμένο (ZIP) αρχείο, που να περιέχει όλα τα αρχεία του πηγαίου κώδικα που δημιουργήσατε για τις ανάγκες της εργασίας σας. Χρησιμοποιήστε την εξής ονοματολογία για το ZIP αρχείο που θα παραδώσετε: **MYE047_LAB2_<YOUR AM>_SOURCE-FILES.zip**.

Επειδή μια MS FORM δεν επιτρέπει τη μεταφόρτωση (upload) ενός ZIP αρχείου, μετονομάστε το αρχείο ZIP που δημιουργήσατε ως εξής (ώστε να είναι εφικτή η μεταφόρτωσή του στη φόρμα υποβολής): **MYE047_LAB2_<YOUR AM>_SOURCE-FILES_zip.pptx**

ΠΡΟΣΟΧΗ: Δε χρειάζεται να συμπεριλάβετε τα (ούτως ή άλλως πολύ μεγάλα για να χωρέσουν) αρχεία δεδομένων της εργασίας. Ζητείται ΜΟΝΟ ο δικός σας κώδικας.

(β) ΠΕΙΡΑΜΑΤΙΚΕΣ ΜΕΤΡΗΣΕΙΣ: Ένα αρχείο XLSX όπου θα περιλαμβάνονται (κατ' ελάχιστον) τα πειραματικά δεδομένα που πήρατε από τις μετρήσεις σας, για τα διαφορετικά πειράματα που κάνατε (ανάλογα με τις επιλογές που δώσατε).

(γ) ΓΡΑΠΤΗ ΑΝΑΦΟΡΑ: Τη γραπτή αναφορά σας (σε DOCX ή/και σε PDF μορφή) για την εργαστηριακή άσκηση MYE047/LAB-2. Χρησιμοποιήστε την εξής ονοματολογία, για το αρχείο πειραματικών μετρήσεων που θα παραδώσετε: **MYE047_LAB2_<YOUR AM>_REPORT.docx** ή **MYE047_LAB2_<YOUR AM>_REPORT.pdf**

Στη γραπτή αναφορά σας θα περιγράφετε αναλυτικά τις μεθόδους που υλοποιήσατε και, κυρίως, θα παραθέτετε όλα τα πειραματικά αποτελέσματα συνοδευόμενα με τον κατάλληλο σχολιασμό. Θα αξιολογηθεί, πέρα από την πληρότητα και την ορθότητα των πειραματικών μετρήσεων, και η επαρκής εξήγηση των μετρήσεων που πήρατε. Π.χ., είναι τα πράγματα όπως τα περιμένατε? Εμφανίστηκε κάποια απροσδόκητη μέτρηση, κι αν ναι, πού μπορεί να οφείλεται αυτό?

Η παράδοση της εργασίας θα γίνει μέσα από τη φόρμα υποβολής με το όνομα

2021-22_CSE/MYE047 : : LAB-2 SUBMISSION FORM...

που θα βρείτε αναρτημένη στο κανάλι LAB-2 στην MS-TEAMS ομάδα του μαθήματος. Ο κώδικάς σας θα αποσταλεί, πάντα εμπρόθεσμα, μέσω ηλεκτρονικού ταχυδρομείου στο kontog@uoi.gr.